[MVA] - Rapport de stage

Mathis Reymond

April 2024

1 Introduction

Continual learning is a machine learning paradigm that focuses on the ability of a model to learn and adapt to new information over time, without forgetting previously learned knowledge. It is particularly relevant in scenarios where data is non-stationary or arrives in a sequential manner. By leveraging techniques such as online learning, regularization, and knowledge distillation, continual learning enables models to continuously update their knowledge and improve performance as new data becomes available.

1.1 Initial project

1.1.1 Project statement

In this context, the aim of the project was initialy stated as follows:

> *Catastrophic forgetting is a major challenge in continual learning systems, where a neural network needs to adapt to new tasks that differ from what it has previously learned. To tackle this, experts have devised strategies to either keep the neural network's critical weights intact or use generative models to refresh the network's memory of past tasks. However, these methods aren't perfect for hardware implementation due to their additional memory demands or the need for a separate network that constantly rehearses past information.*
>
> *Meanwhile, we understand that a system's architecture significantly impacts its ability to generalize knowledge. Enter Gradmax [4], an innovative algorithm that simultaneously optimizes a network's structure and weights, and smartly allocates neurons for new tasks without disrupting existing knowledge. This method is memory and computation efficient, making it suitable for hardware applications. But it does require hardware capable of dynamically adjusting its architecture to integrate new neurons as needed. Our recent work has led to a flexible hardware architecture, Mosaic [1], with multiple cores that can rearrange their connections in real-time.*
>
> *The aim of this project is to explore the integration of Gradmax with our dynamic 'Mosaic' architecture. We plan to approach this by factoring hardware limitations into the optimization process. In addition, we're considering the development of a gating network within this architecture that determines the optimal timing for assigning tasks to newly added neurons. This could leverage cutting-edge techniques like Mixture of Expert models [2].*

Let's dissect this project statement to understand the main objectives and challenges it presents.

Computational costs. The project aims to address the computational costs associated with continual learning methods. By leveraging Gradmax and Mosaic, the project seeks to reduce these costs and make continual learning less expensive.

Continual learning approach. Litterature categorizes continual learning into three main categories: rehearsal, regularization, and parameter isolation.

1.1.2 Major issues

However, in the very beginning of the project, we faced two major issues. The first one is a consequence of Gradmax paper misleading introduction. Indeed, in the very first paragraph of the paper, the authors introduces :

> *Searching for the best architecture for a given task is an active research area with diverse approaches.[...] Most of these approaches are costly, as they require large search spaces or large architectures to start with. In this work we consider an alternative approach.*

The genuine expectation regarding the work following such an introduction is that the authors will provide an inovative method that performs factorize architecture search in the optimization process of learning a task. We also expect that, contrary to existing methods in this domain, this method does not require to train numerous architectures of the final size, which is computationally expensive.

Contrary to expectations, GradMax falls short at enabling neural architecture optimization. It consists in growing a network from a root architecture to a predetermined target architecture, by adding neurons to it through training epochs. It focuses on the initialization of added neurons (the *how*), neglecting the crucial questions of *when* and *where* to add them, which are deferred for future research. In the paper, the decisions regarding *where* and *when* to add neurons are predetermined and take the form of a schedule established. Consequently, it's challenging to conclude that the GradMax framework effectively enables architecture optimization for growing artificial neural networks.

This introdcutory statement is also misleading in the sense that authors do not consider the computational cost of their method in the rest of the paper. This is a major issue as the project was initially designed to reduce the computational cost of continual learning methods.

1.1.3 Additional hurdles

1.2 Revised project

References

[1] Thomas Dalgaty et al. "Mosaic: in-memory computing and routing for small-world spike-based neuromorphic systems". In: *Nature Communications* 15.1 (Jan. 2024). ISSN: 2041-1723. DOI: `10.1038/s41467-023-44365-x`. URL: `http://dx.doi.org/10.1038/s41467-023-44365-x`.

[2] David Eigen, Marc'Aurelio Ranzato, and Ilya Sutskever. *Learning Factored Representations in a Deep Mixture of Experts.* 2014. arXiv: `1312.4314 [cs.LG]`.

[3] Jason K. Eshraghian et al. *Training Spiking Neural Networks Using Lessons From Deep Learning.* 2023. arXiv: `2109.12894 [cs.NE]`.

[4] Utku Evci et al. *GradMax: Growing Neural Networks using Gradient Information.* 2022. arXiv: `2201.05125 [cs.LG]`.

[5] Ziming Liu et al. *Growing Brains: Co-emergence of Anatomical and Functional Modularity in Recurrent Neural Networks.* 2023. arXiv: `2310.07711 [q-bio.NC]`.

[6] Javier Lopez Randulfe and Leon Bonde Larsen. *A multi-agent model for growing spiking neural networks.* 2020. arXiv: `2010.15045 [cs.NE]`.

[7] Xin Yuan, Pedro Savarese, and Michael Maire. *Accelerated Training via Incrementally Growing Neural Networks using Variance Transfer and Learning Rate Adaptation.* 2023. arXiv: `2306.12700 [cs.LG]`.